实验一　选择器

2020 年秋季学期

杨飞洋 191220138

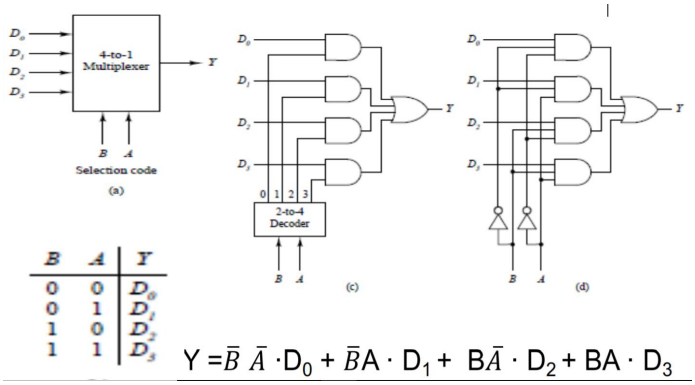
目录

1. 实验目的：

了解 Verilog 语言中的 always 语句块、if-else 语句和 case 语句的使用等。
用 case 语句实现一个 2 位 4 选 1 的选择器，如图 1 5 所示，选择器有 5 个 2 位输入端，分别为 X0, X1, X2, X3 和 Y，输出端为 F；X0, X1, X2, X3 是四个 2 位的输入变量。输出 F 端受控制端 Y 的控制，选择其中的一个 X 输出，当 Y =00 时，输出端输出 X0，即 F = X0；当 Y = 01 时，输出端输出 X1，即 F = X1；以此类推。
选择开发板上的 SW0 和 SW1 作为控制端 Y，SW2—SW9 作为四个两位数据输入端 X0–X3，将两位的输出端 F 接到发光二极管 LEDR0 和 LEDR1 上显示输出，完成设计，对自己的设计进行功能仿真，并下载到开发板上验证电路性能。

2. 实验原理（知识背景）

选择器是数字逻辑系统的常用电路，是组合逻辑电路中的主要组成元件之一，它是由多路数据输入、一位或多位的选择控制端，和一路数据输出所组成的。多路选择器从多路输入中，选取其中一路将其传送到输出端，由选择控制信号决定输出的是第几路输入信号。
下图是四选一多路选择器的真值表及其电路原理图。



B	A	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Y = \bar{B}\,\bar{A}\cdot D_0 + \bar{B}A\cdot D_1 + B\bar{A}\cdot D_2 + BA\cdot D_3$$

3. 实验环境/器材

系统：windows10
开发软件：Quartus 17.1 Lite
开发板：DE10 Standard
仿真环境：ModelSim
芯片：Cyclone V ， 5CSXFC6D6

4.设计思路

现在是将 4 个两位数据作为输入端，设计真值表为：

SW[0]	SW[1]	LEDR[0]	LEDR[1]
0	0	SW[2]	SW[3]
1	0	SW[4]	SW[5]
0	1	SW[6]	SW[7]
1	1	SW[8]	SW[9]

相当于将用两个四选一多路选择器，共用一个数据选择端，将 SW[2],SW[4],SW[6],SW[8]和 SW[3],SW[5],SW[7],SW[9]进行分组，分别得出结果输出到 LEDR[0]和 LEDR[1]上。

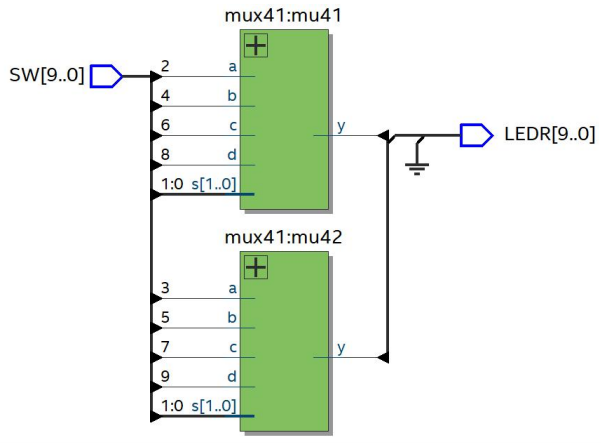
module 代码：

```
module mux41(a,b,c,d,s,y);
    input a,b,c,d;
    input [1:0] s;
    output reg y;

    always @ (s or a or b or c or d)
        case (s)
            0: y = a;
            1: y = b;
            2: y = c;
            3: y = d;
            default: y = 1'b0;
        endcase

endmodule
```

顶层文件调用：

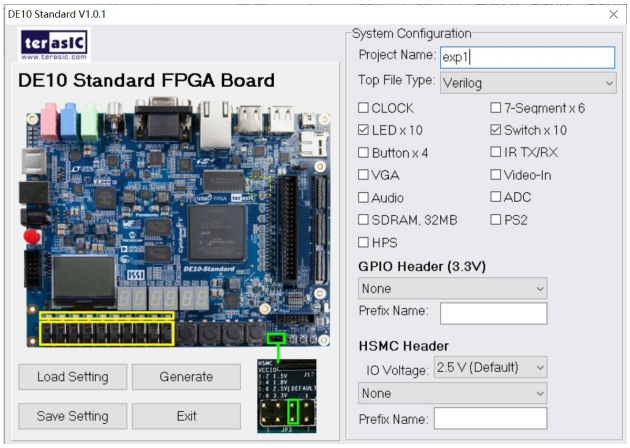
```
mux41 mu41(SW[2],SW[4],SW[6],SW[8],SW[1:0],LEDR[0]);
mux41 mu42(SW[3],SW[5],SW[7],SW[9],SW[1:0],LEDR[1]);
```

RTL viewer 电路图：



5.实验步骤

一．利用 systembuilder 建立工程



二．编写 Verilog 代码

在工程目录打开 qpf 文件 exp1.qpf

新建一个.v 文件，写一下四选一多路选择器的模块 `module mux41(a,b,c,d,s,y);`

再将该文件名更改为 mux41.v,（文件名与模块名一致）

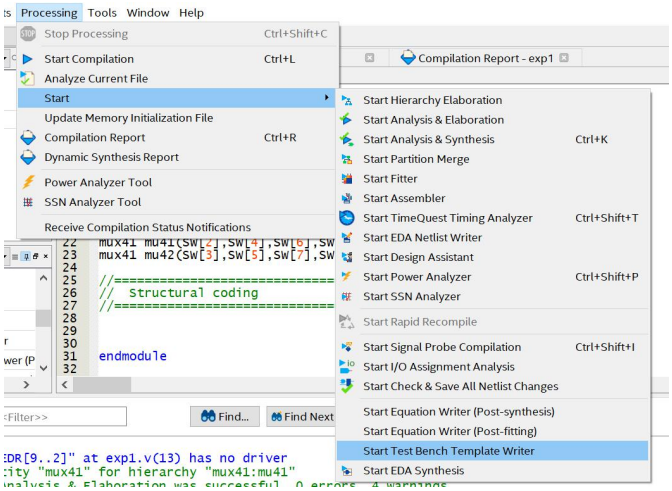
在顶层文件中根据需要调用 mux41 模块

```
mux41 mu41(SW[2],SW[4],SW[6],SW[8],SW[1:0],LEDR[0]);
mux41 mu42(SW[3],SW[5],SW[7],SW[9],SW[1:0],LEDR[1]);
```

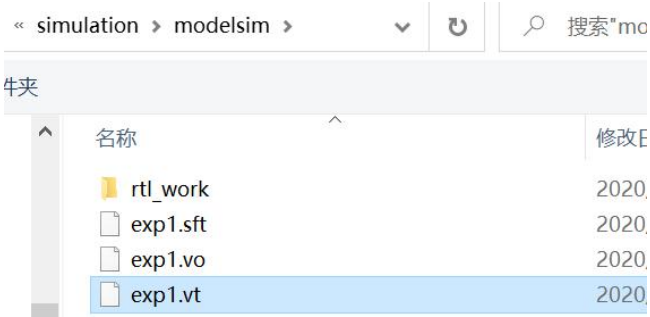
进行编译调试，直至成功。

三．编写 test bench

利用 processing 中的 start test bench template writer 让系统自动生成 test bench 测试文件



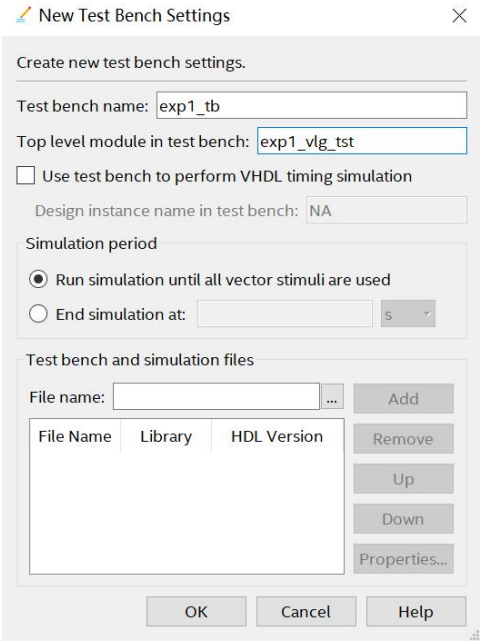
在 simulation 中的 modelsim 文件中打开.vt 文件



修改仿真时间单位，`timescale 10_ns/ 1_ps
在 initial 下方 begin 和 end 之间写入测试数据
为避免风险，将 always 语句注释。

四．设置 test bench

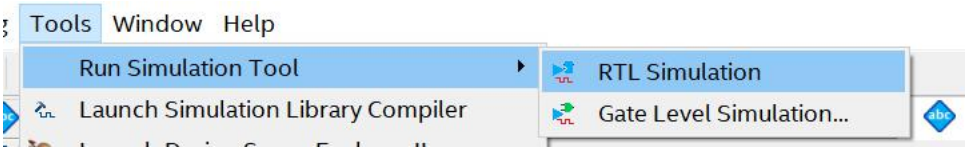
在 assignments->settings->simulation->compile test bench 中设置 test bench，new 一个 test bench，注意将顶层实体修改为 exp1_vlg_tst



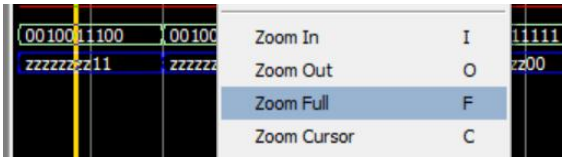
再将 simulation->modelsim 中的.vt 文件 add 进去

五．仿真测试

利用 RTL Simulation 调用 modelsim



在波形图中 zoom full 就可以看到测试结果

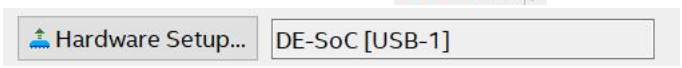


六．利用 fpga 测试

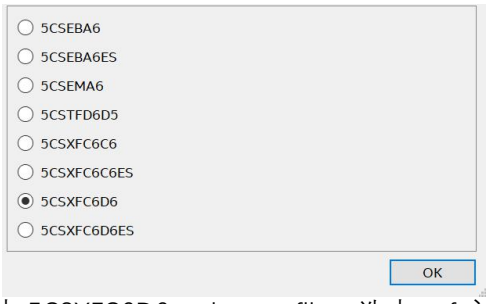
检查工程目录下有 sof 文件

将 fpga 充电，usb 接电脑，点击 programmer 按钮

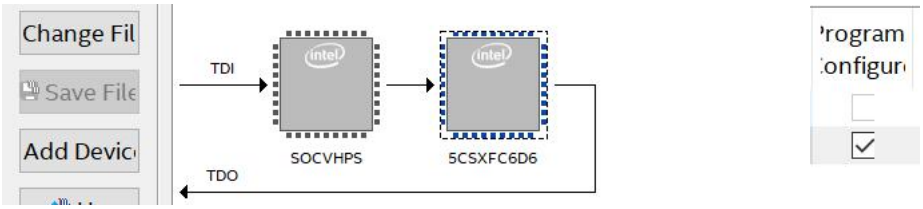
hardware 选择 usb



select device 选择倒数第二个



选中 5CSXFC6D6，change file，选中.sof 文件，接着在 program 上打勾，在点击 start 开始测试



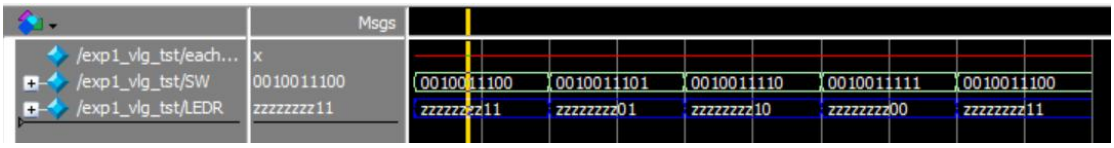
6.测试方法

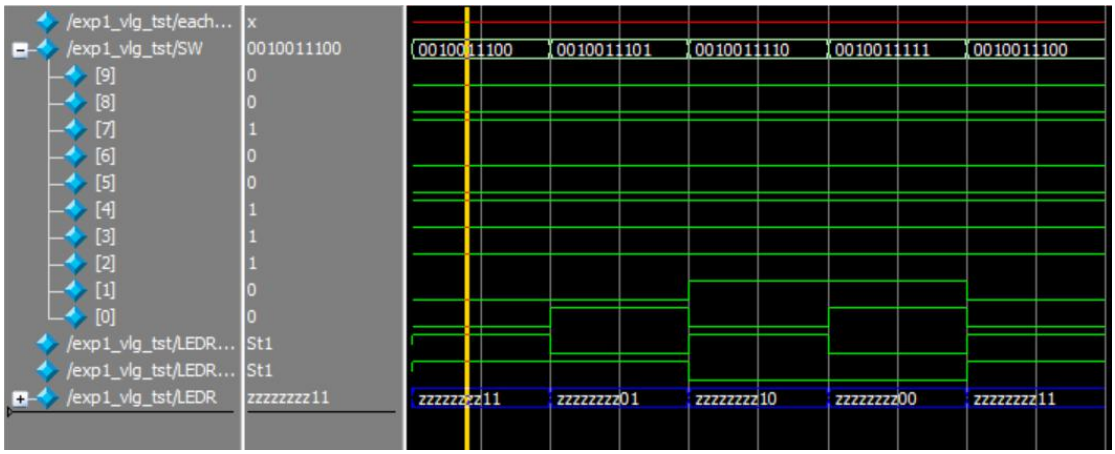
分四个状态，分为(SW[0],SW[1]) = (0,0) , (1,0) , (0,1) , (1,1)，将 SW[2:9]赋值为 11100100，这样不同的状态下输出也都是不同的。下图是测试文件代码：

```
initial
begin
// code that executes only once
// insert code here --> begin
SW[9:0] = 10'b0010011100;#20;
SW[9:0] = 10'b0010011101;#20;
SW[9:0] = 10'b0010011110;#20;
SW[9:0] = 10'b0010011111;#20;
SW[9:0] = 10'b0010011100;#20;
// --> end
$display("Running testbench");
end
```

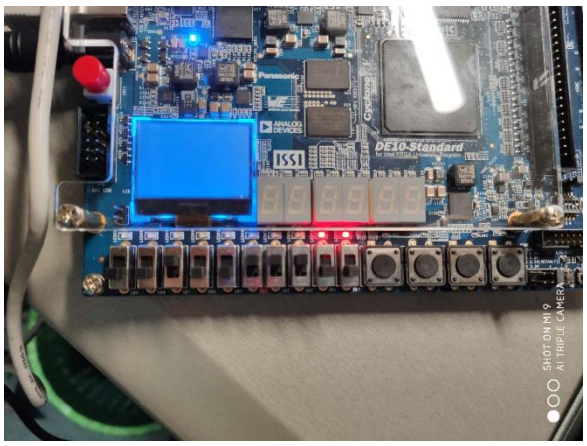
7.实验结果

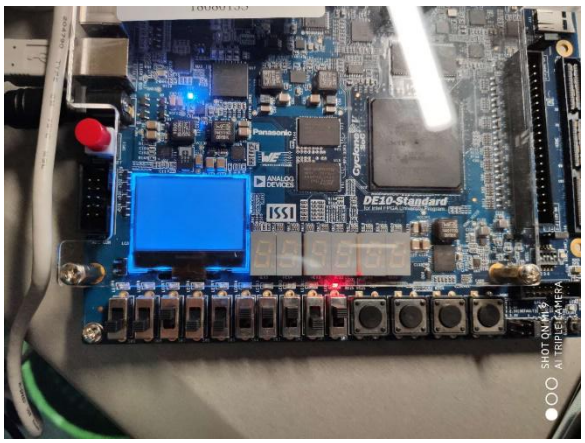
仿真结果：

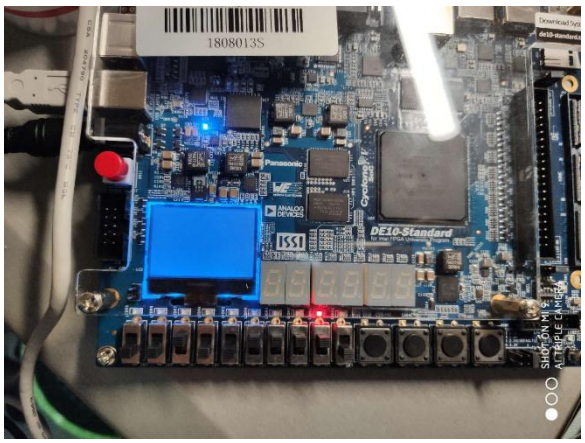


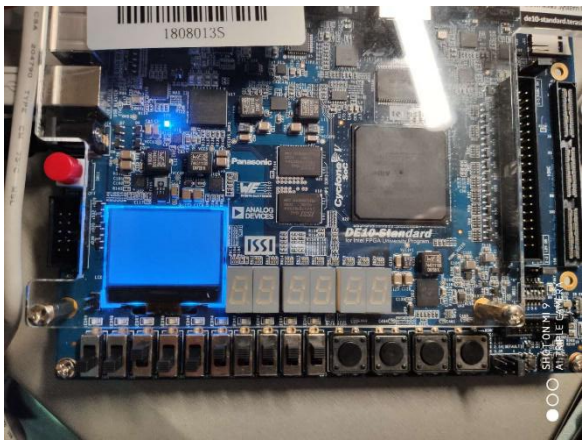


下载运行结果：









8.实验中遇到的问题

一．安装时多次遇到 error，经过总结，有安装路径存在中文字符的错误、有安装时内存超载的错误。

二．进行仿真测试时，忘记将顶层实体修改为 exp1_vlg_tst，导致问题。

三．Verilog 的语法规范还不是特别熟悉，经常会编译错误。

四．LEDR[0]、LEDR[1]由于没有谨慎考虑，经常会输出相反。

五．在找.vt 测试文件时一直找不到，因为没有开在全部文件中搜索

9.实验得到的启示

需要仔细阅读编译信息，方便找到 bug。
要按照流程按部就班的做，不然一步错全部错。
需要理解 wire 和 reg 的区别。

10.意见和建议

希望可以提供一个现成的安装包，那个网站真的很难进。
希望软件可以优化一下编译速度。